AFRL-RI-RS-TR-2009-141
Final Technical Report
May 2009

DESIGN ISSUES OF A MIND SNAPS IMPLEMENTATION

SET Corporation

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2009-141 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/	/s/
JOHN SPINA	JOSEPH CAMERA, Chief
Work Unit Manager	Information & Intelligence Exploitation Division
	Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
MAY 09	Final	Jul 08 – Jan 09

4. TITLE AND SUBTITLE
DESIGN ISSUES OF A MIND SNAPS IMPLEMENTATION

5a. CONTRACT NUMBER FA8750-08-C-0212
5b. GRANT NUMBER N/A
5c. PROGRAM ELEMENT NUMBER N/A

6. AUTHOR(S)
Rafael Alonso and Hua Li

5d. PROJECT NUMBER R665
5e. TASK NUMBER 08
5f. WORK UNIT NUMBER 04

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
SET Corp.
1005 North Glebe Road, Suite 400
Arlington, VA 22201-5758

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFRL/RIED
525 Brooks Rd.
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
N/A

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2009-141

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 88ABW-2009-2069

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The intelligence analysis process consists of a complex, iterative, highly-branched sequence of information gathering and processing steps. Throughout the process, the analyst needs to refer to the hypotheses and information he or she was manipulating at previous points in the process. A key enabler for the analytic process would be to provide analysts with semantic bookmarks, i.e., a mechanism that would allow them to return to a particular point in the analysis and recreate the complete context they had at that time. A key element of the programmatic vision for the next phase of IARPA¹s A-SpaceX program is to define, explore, and implement such semantic bookmarks, which are referred to as Mind Snaps. We have completed a short (approximately six months) seedling to explore the Mind Snaps concept, focusing on answering several research questions. The answers to these questions will serve not only to better define and understand Mind Snaps, but also guide possible programmatic goals.

15. SUBJECT TERMS
Information gathering, Mind snaps, Intelligence analysis

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON / 19b. TELEPHONE NUMBER (*Include area code*)
U	U	U	UU	16	John Spina / N/A

Table of Contents

List of Figures

List of Tables

Final Seedling Report on

Design Issues of a Mind Snaps Implementation

Dr. Rafael Alonso
Dr. Hua Li

{ralonso, hli}@SETcorp.com

SET Corporation

March 26, 2009

1 Mind Snaps Background

The intelligence analysis process consists of a complex, iterative, highly-branched sequence of information gathering and processing steps. Throughout the process, the analyst needs to refer to the hypotheses and information he or she was manipulating at previous points in the process. A key enabler for the analytic process would be to provide analysts with *semantic bookmarks*, i.e., a mechanism that would allow them to return to a particular point in the analysis and recreate the complete *context* they had at that time. A key element of the programmatic vision for the next phase of IARPA's A-SpaceX program is to define, explore, and implement such semantic bookmarks, which are referred to as *Mind Snaps.* We have completed a short (approximately 6 months) seedling to explore the Mind Snaps concept, focusing on answering several research questions. The answers to these questions will serve not only to better define and understand Mind Snaps, but also guide possible programmatic goals in a future IARPA program.

2 Mind Snaps Meetings and Workshops

As part of the seedling work, SET also helped plan and lead several workshops, at SET's Greenbelt office, at SET's Ballston office, and in Denver, at the University of Denver's facilities. The workshops included several well-known industrial researchers and academics, and the meeting results were captured in PowerPoint presentations. These presentations as well as each individual researcher's presentations (from SET and elsewhere) were delivered to the seedling program manager, Dr. Jeff Morrison (IARPA).

3 Using Mind Snaps

Below is a simplified Concept of Operations we developed for one of the Mind Snaps (MS) workshops. Imagine an analyst working on a given tasking:

1. After several months of work, the analyst needs to revisit old work (maybe new data came in or a line of hypothesis proved futile but vague recollection of something useful in past)
2. The analyst needs to move in a virtual world that represents his or her past work. The analyst can browse past actions and recognize enough milestones such that meaningful (episodic, time, etc,) MS queries can be asked
3. The analyst asks for a MS using some query language or more likely a visual interface

4. The system reasonably quickly shows the analyst a picture (in the virtual world) of what his context looked at the time of the MS
5. Files, documents, etc., are open to the right place, tools are loaded with the right data, etc.
6. Any changed information is highlighted
7. The analyst can quickly remember the “big picture” of what he or she was thinking about
8. The analyst refines his or her memories and is back to the mental and cyber context he or she had months before

In supporting this concept, three very difficult technical issues need to be addressed. The first, is the careful disentangling of the analyst’s micro-contexts. To clarify, the analyst engages in multiple activities during a given time period, e.g., a morning. Each of those activities belongs to a different task, and thus that work needs to be coupled with other task-related work. Separating actions into multiple micro-contexts is a key technical hurdle that must be addressed so as to create an effective Mind Snaps mechanism. The second technical difficulty lies in helping the analyst visualize the entirety of his or her analytic mind-space so he or she can ask for the correct mind snap. The third technical difficulty lies in, once the right mind snap is found, helping the analyst to gestalt or comprehend the work that he or she once understood well.

While all three technical issues mentioned are worthy of further study, the bulk of the work in our seedling was devoted to exploring the first technical issue above (separating micro-contexts), and determining the feasibility of tackling it in an IARPA program. The sections below describe our technical approach, and why we believe our results validate the concept.

4 Mind Snap Frequency

A key question for Mind Snaps is how often they should be created. (One may also ask whether Mind Snaps should be discrete or continuous, but that issue will not be addressed here.) The answer is that they should be created often enough that shifts in analytic tasks can be captured but not so often as to create a burden on the computational resources. The key challenge here is to identify such shifts, whether they result from the analyst exploring alternative hypotheses or from being engaged in multiple simultaneous tasks. We explain our approach for identifying context switches below.

4.1 Preliminary Experiment Exploring Mind Snap Frequency

The minimum time periods between contextual shifts depend on the least amount of analytic activity required to construct a thematically meaningful context. We call this quantity the minimal meaningful work (MMW). In a preliminary experiment, we attempted to assess the MMW by measuring our ability to distinguish analysts working on different tasks as a function of the amount of analytic activity. The smaller amount of analytic activity required for successful separation of tasks, the smaller the MMW, and the more often the Mind Snaps should be created.

Analyst activities can be captured and persisted as analysis log events (ALEs). The ALE specification was developed by IARPA’s CASE program[1], a systematic taxonomy for describing user events commonly occurring in information systems. The ALE specification includes four high-level event classes: information processing events, social network event, workstation event,

[1] IARPA, CASE Analysis Log Service Specification, version 3.0, June, 2008.

and network environment event. Information processing ALEs include search (e.g. query), access (e.g., browsing), retain (e.g., cut/paste), assess (e.g., rating), annotate (e.g., bookmark), discard (e.g., delete a document), etc.

We measure the ability to distinguish analysts working on different tasks by identifying the virtual interest group (VIG) for each user. Obviously, a user's VIG should contain users working on the same task. We have used Model-based Algorithm for VIG Identification to compute VIG in this experiment. This algorithm identifies VIG's by directly comparing the similarities of user models built with data segments from each user.

4.1.1 Data

The data set for this experiment comes from the Glass Box data maintained by NIST. It consists of 8 users. The task, dates of analysis activity and number of ALEs are detailed below.

- Ccgbuser4
  - 100401-Syria Reaction
  - 11/14 – 11/23/2005
  - 588 ALEs
- Ccgbuser7
  - 100701-FSU Biotech Council
  - 11/14 – 11/22/2005
  - 1107 ALEs
- lc1 – lc6
  - lc week 2
  - 11/14 – 11/18/2005
  - ALEs: lc6=487, lc5=311, lc4=409, lc3=304, lc2=318, lc1=135

The task information is available in the data for user ccgbuser4 and ccgbuser7, but is intentionally ignored during the VIG computation. The ground truth for the user groupings provided by NIST is used for calculating the precision of the VIGs.

4.1.2 Setup

Each user has about one-week worth of data. For each working day of each user, we take the initial segment of N ALEs to build a user model. N is set to 3, 5, 10, 20, 30, 50, or 100. For each choice of N, we build a total of 38 daily user models. These daily models are used for VIG computation. The precision of the VIG is calculated as the ratio of percent of correct user models.

4.1.3 Results

The results are shown in **Figure 9**. We make the following findings:

- The VIG precision improves as more ALEs are used.
- Over 70% precision can be achieved with as little as 3 ALEs whereas close to perfect with 100 ALEs.
- By using 10 to 30 ALEs, 85% to 95% of precision can be achieved.

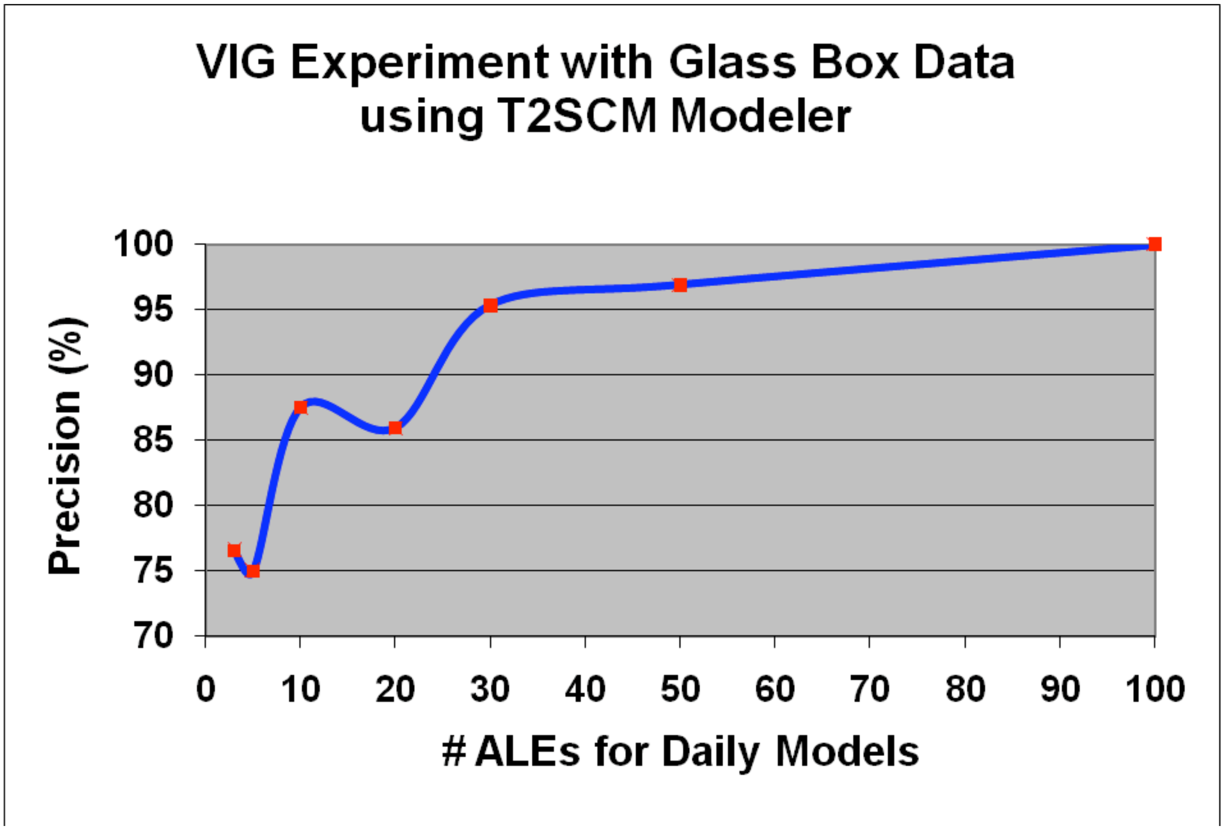

Figure 9. Graph shows the VIG identification performance as function of number of ALEs.

4.2 Conclusions from the Preliminary Experiment

The results from the experiment suggest that the MMW of 10 to 30 ALEs offers 85% to 95% precision in the ability to distinguish analysts working on different tasks. According to NIST and our own observation, on average, analysts generate 2 ALEs every minute. Therefore, 10 to 30 ALEs correspond to about 5 to 15 minutes of analytic activity. This suggests that the interval for taking Mind Snaps should be between 5 to 15 minutes.

5 Separating Contexts

The most technically difficult task in the proposed seedling is the teasing apart of overlapping contexts in the analyst workload. As described above, separating contexts is central to deciding the Mind Snap frequency, as well as identifying which entities (documents, queries, etc.) should belong to the current Mind Snap. There are two central research issues to be tackled in this task. First, we need to consider how to identify shifts in analyst focus. Second, for a given approach, we need to determine how well it performs, both in terms of correctness, as well as in terms of the smallest size contexts that can still be separated.

5.1 An Algorithm for Separating Context based on User Modeling (ASCUM)

This algorithm aims to separate entwined contexts with the use of user models. The user models will be built using the Reinforcement and Aging Modeling Algorithm (RAMA). RAMA

combines reinforcement learning with information aging. Positive events express user's interests and increases the importance of the topics contained in them. On the other hand, negative events imply user's disinterest and thus decrease the importance of the contained topics. We model the topics implied in the ALEs using a information object modeler, which is a NLP tool that models the textual content associated with an ALE. The T2SCM (Text to Specialized Concept Map) modeler extracts typed concepts and relations from English text and generates a specialized concept map represented as an XML document. The TF (Term Frequency) modeler extracts the normalized term frequencies of terms in the document.

The idea behind ASCUM is complex in its realization but relatively straightforward to describe. For several analysts, we will identify all the documents they have accessed during a long activity trace. We will use TF or T2SCM to extract the concepts and relations from the documents. We then create dynamically-adaptive analysts models that can capture at each point in time the analysts' thematic interests, as well as their relative level of interest (the resulting model can be thought of as a graph where nodes are clusters, edges the information distance among the clusters, and the interest levels as annotations on the nodes). As the analyst models evolve with the analyst's activities, we will define a new context when the model changes "enough." We will determine what constitutes a sufficiently large distance to identify most shifts in analytic focus by running the algorithms on realistic analysis data.

5.1.1 User Models used in ASCUM

- **Current short-term model model (CSM)**
    - o Captures latest user interests
    - o built with latest N=10 ALEs
- **Last short-term model (LSM)**
    - o Captures recent short-term user interests
    - o built with N ALEs preceding those used for CSM
- **Lifetime model (LM)**
    - o Captures user interests from the start of the system
    - o built with all available ALEs except those used by the CSM
- **Context Model**
    - – A user model that captures user's interests related to a goal or task

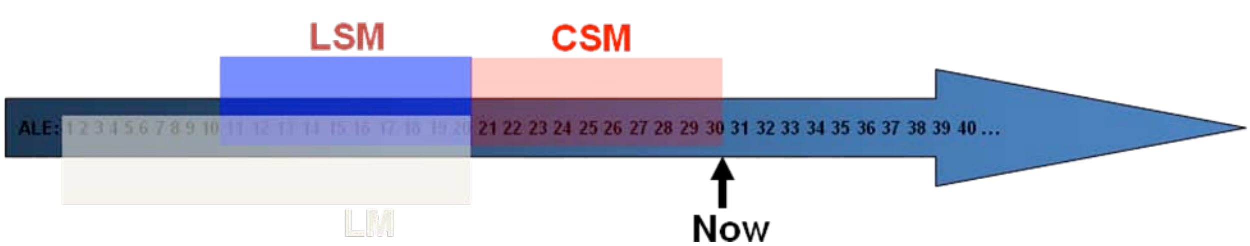

Figure 10. An ALE data stream showing CSM, LSM, and LM.

5.1.2 Context Management

- Input: continuous ALE stream
- Build LM, CSM, and LSM with the first N ALEs
- Saving CSM as the current context
- Loop

- Retrieve the next N ALEs
- Build the new CSM
- Perform context shift detection
- Identify contexts
- Save contexts
- Update the LM with the N ALEs
- Make the CSM the new LSM

5.1.3 Context Shift Detection

Context switch occurs if the current model differs from either the lifetime model or the last short-term model. Two models differ if their similarity is below preset threshold.

Table 2 - Context shift detection matrix. Note the symbols: "=": similar, "!=" different.

	CSM = LSM	**CSM != LSM**
CSM = LM	**No switch**	**Switch to existing context – merge CSM with existing context**
CSM != LM	**Context drift (transition to new context)**	**Switch to new context – save CSM as new context**

5.1.4 Context Identification

The goal here is to determine what context the current model (i.e., CSM) belongs to. To achieve this, we first compare current model with saved context models. We then select the most similar context model as the context the CSM belongs to.

5.1.5 Saving Contexts

- In the cases of "No switch" and "Context drift", no actions taken.
- In all cases, the CSM will be merged with the LM.
- Merging CSM with a model means updating the latter with the ALEs that the CSM is built on. Alternatively, we can also merge by directly incorporating CSM concepts into the target.

5.2 Performance Evaluation of ASCUM

To test how well our approach works, we will obtain workflow trace segments from several analysts and integrate the segments into a single trace. Then we will determine the percentage of trace segments that can be correctly separated. We will then vary the size of the trace segments, from relatively large ones to ones so small that cannot be usefully separated.

5.2.1 Experimental Setup

- **Data**
  - CASE evaluation data archive from the NIST ALS
  - 28 users
  - 4029 ALEs

- **ALE Stream**
  - o Linearly combine the data of all users as a continuous stream of ALEs.
  - o The users are temporally separate in the stream.
- **Context Shift Detection Parameters**
  - o Different window sizes are used: 10 and 30 ALEs
  - o Default thresholds for 10- and 30-ALE window are 0.9 and 0.85 but can be adjusted in post-processing.
  - – Different object modelers are used: TF and T2SCM.

5.2.2 A Random Detector

To gauge the performance of our context shift detection algorithm, we have constructed binomial-distribution-based random detectors with the following constraints:

- We have 28 users in sequence. Assume that each user works on a different task from the next in line, we have maximal possible correct context switches of 27.
- We have a total of 4029 ALEs. For a given window size (30 or 10 ALEs), we have fixed number of windows for switch detection.

5.2.2.1 A Random Detector with a Window Size of 30 ALE

With 30 ALEs in a window, we have a total of 134 windows. Given 27 possible correct switches, the probability for a success (i.e. correctly identify a context switch) in a given trial is 27/134=0.2. Assuming binomial distribution, the probability to get 19 successes out of 26 random trials is (**Figure 11**):

- 19/26: p=8.2E-9

Similarly we have:

- 18/44: p=0.00089

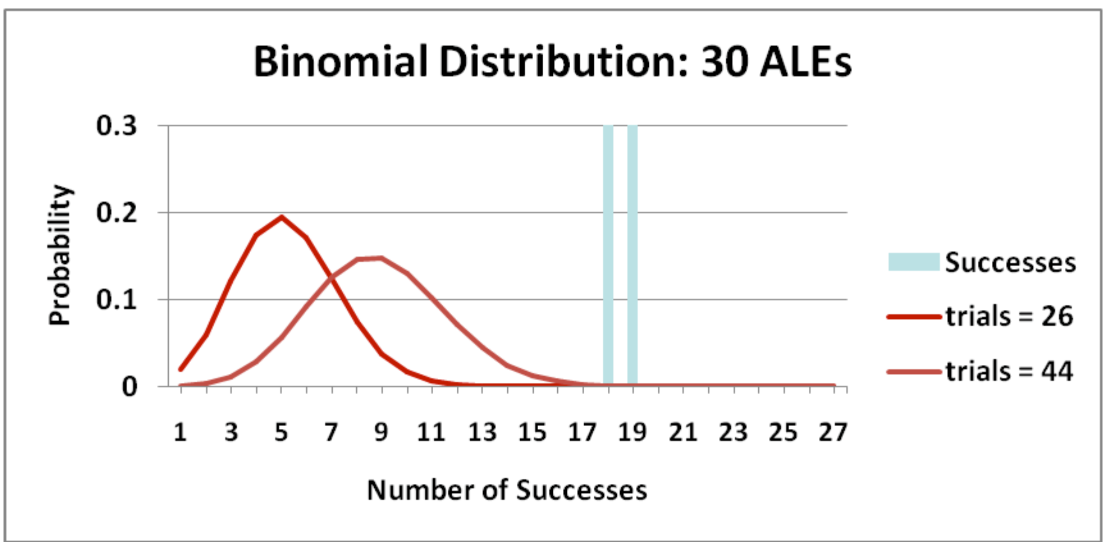

Figure 11. A binomial-distribution-based random detector for context shift for window size at 30 ALEs.

5.2.2.2 A Random Detector with a Window Size of 10 ALE

With 10 ALE windows, we have the following results:

- 402 windows

- Probability of a success: p = 27/402 = 0.067
- 19/53: p=5.8E-10
- 22/124: p=1.8E-5

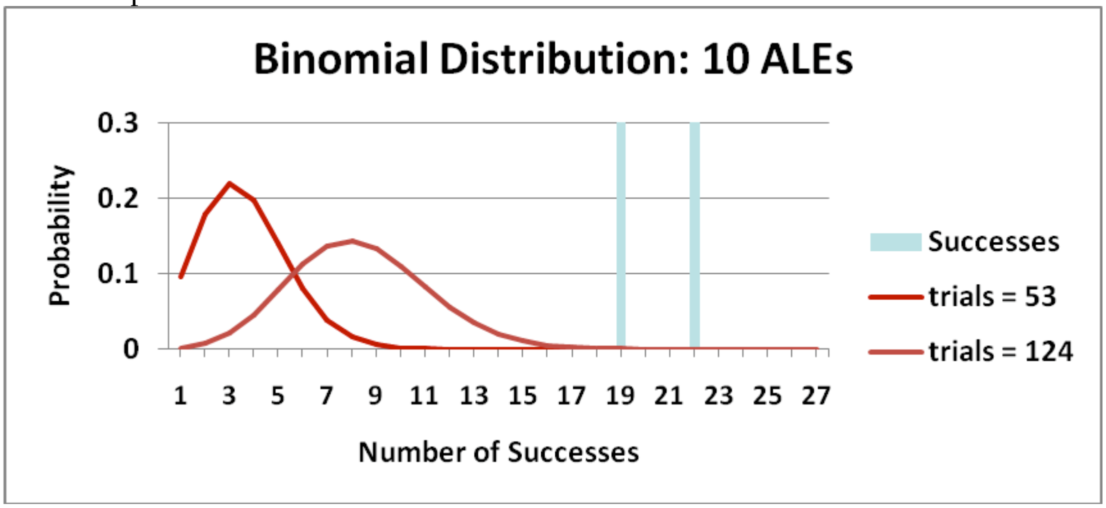

Figure 12. A binomial-distribution-based random detector for context shift for window size at 10 ALEs.

5.2.3 Results

The performance of context shift detection is impacted by the window size for LSM and CSM, and by the information object modeler used. We also examined the relative usefulness of LSM, LM and context model as marker for context shifts.

5.2.3.1 Effects of Window Size

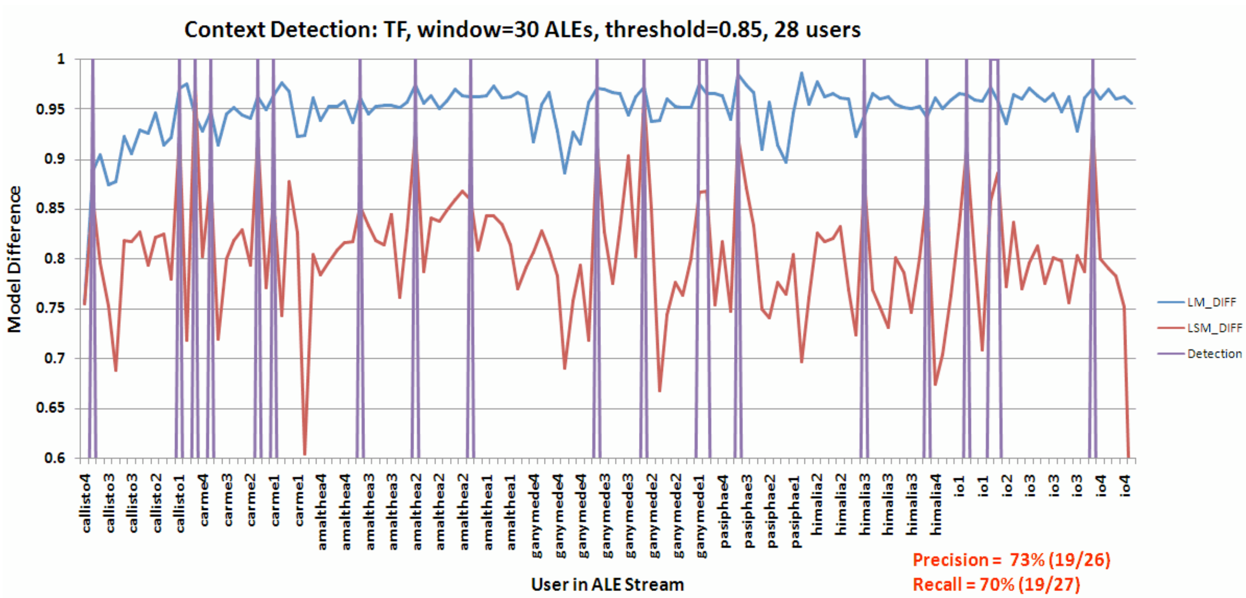

Figure 13. Context shift detection with window size of 30 ALEs, TF modeler, and a threshold of 0.85. Also note that the blue trace is the difference between the CSM and the LM over the course of the ALE data stream.

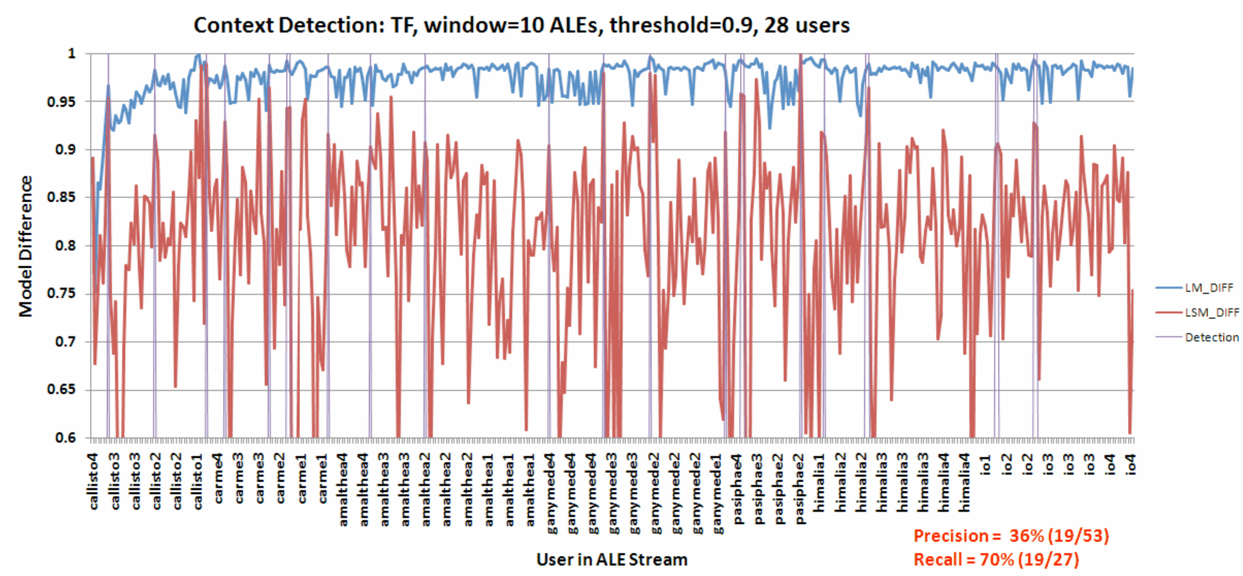

Figure 14. Context shift detection with window size of 10 ALEs, TF modeler, and a threshold of 0.9.

5.2.3.2 Effects of Object Modelers

Two object modelers are used: TF and T2SCM. With window size of 30 ALEs, the performance for TF is shown in **Figure 13** and that for T2SCM is shown in **Figure 15**.

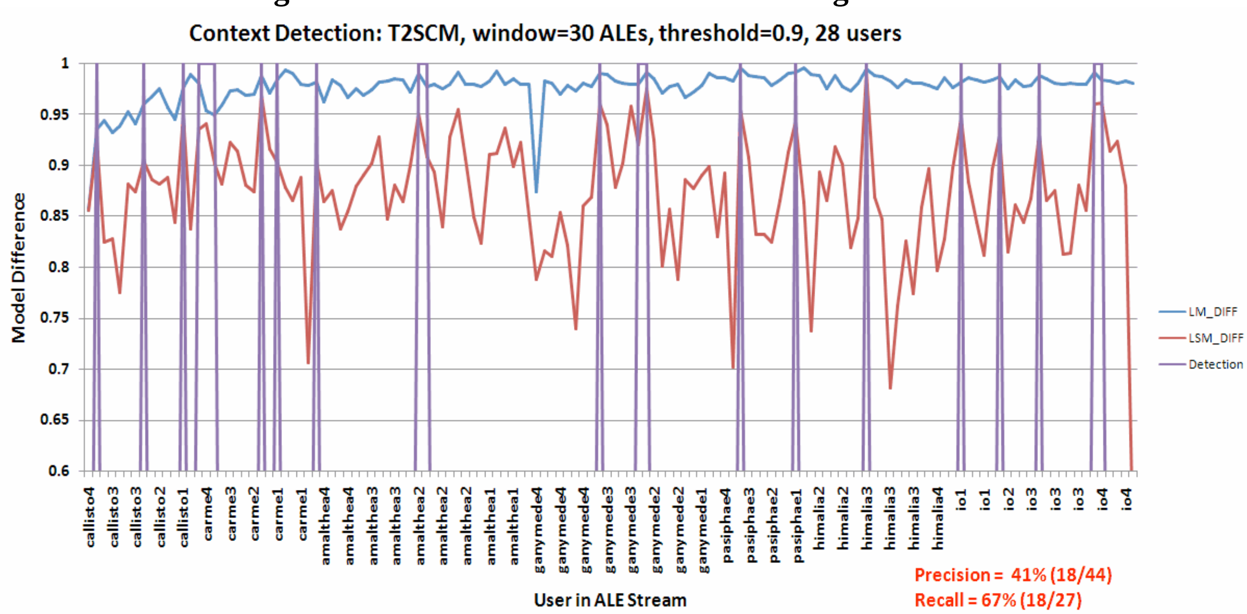

Figure 15. Context shift detection with window size of 30 ALEs, T2SCM modeler, and a threshold of 0.9.

5.2.3.3 Utility of LM and Context Model

We have looked into the usefulness of LM and context model in detecting context switches. We plotted the model difference between LM and CSM (**Figure 13**) and between Context Model and CSM (**Figure 16**). In both cases we also plotted the difference between LSM and CSM. We

observe that LSM is most sensitive in detecting context switches. The context model is less, while the LM is the least sensitive.

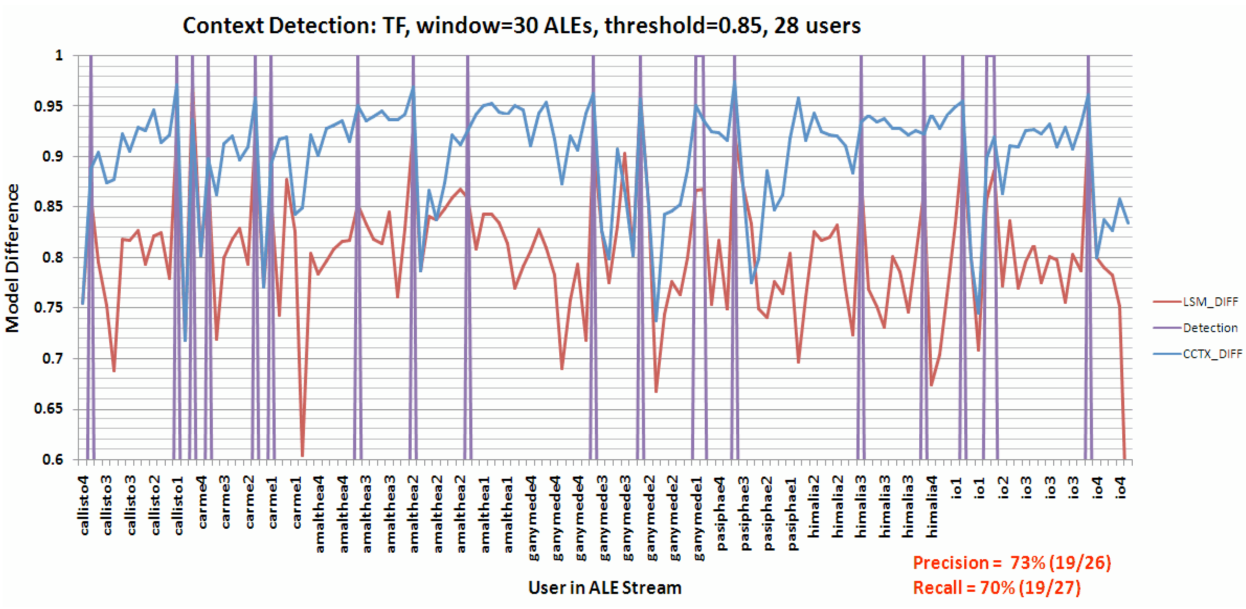

Figure 16. Context model in the context shift detection. The blue trace is the difference between the CSM and the context model over the course of the ALE data stream.

5.2.3.4 Conclusions

- Context switches can be successfully detected using as few as 10 ALE windows with a precision of 36% and a recall of 70%. The precision is significantly better than a random detector at p=5.8E-10.
- Better detection is achieved with larger windows. At window size of 30 ALE, we achieved a precision of 73% (p=8.2E-9) and a recall of 70%.
- Most critical factor for the detection is the comparison between CSM and LSM. The comparison between CSM and LM does not seem to be useful.
- Object modeler TF offers better performance than T2SCM.

5.2.3.5 Next Steps

- Try more window sizes, e.g. 5 ALEs, 20 ALEs, and 60 ALEs to get a performance curve with window size.
- Vary data stream structure. For example, insert one user's ALEs in different parts of the stream and test if the system can pick out the user in those parts.
- Investigate the value of comparison between CSM and the context model in the context switch detection.
- Use different data sets, e.g. the Glass Box data, and the APEX data.

6 Concluding Remarks

Clearly there are several Mind Snap implementation issues not addressed in this seedling, and their study will require a full IARPA project. In addition, we did not address in this seedling other important uses of Mind Snaps. For example, we believe that a Mind Snap can be useful not

only in helping the analyst to remember context, but also in guiding a data gathering tool to fetch documents that are relevant to the analyst's current Mind Snap. Also, Mind Snaps can be used to find mentors for an analyst (individuals whose previous Mind Snaps show them to have delved in the same areas that currently interest the analyst), as well as potential collaborators (by identifying other analysts' Mind Snaps that are similar). However, we believe the completed work clearly shows the feasibility of capturing and disentangling the multiple analytic micro-contexts in which an analyst typically engages, and this was the key technical issue we set out to address. We hope these results will support the formulation of a future Mind Snaps-related IARPA program effort.

7 References

The work we completed in this seedling was documented in monthly reports submitted to the government. The techniques described here are based upon previously completed government sponsored projects, and the results of that work have been documented in the following refereed publications.

Alonso2003a - R. Alonso, J. Bloom, and H. Li. Lessons from the Implementation of an Adaptive Parts Acquisition ePortal. In *Proceedings of 2003 Conference on Information and Knowledge Management.*, 2003.

Alonso2003b - R. Alonso, J. Bloom, H. Li, and C. Basu. An Adaptive Nearest Neighbor Search for a Parts Acquisition ePortal. In *Proceedings of Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, 2003.

Alonso2005 - R. Alonso and H. Li. Model-Guided Information Discovery for Intelligence Analysis. In *Proceedings of Proceedings of CIKM '05*, Bremen, Germany, 2005.